



1501

AF

In re: Application of	Victor Gorelik
Application No.	10/725,116
Filed	12-02-2003
Title	TWISTED SIGNATURE
Examiner	LOUIE, OSCAR A
Art Unit	2136

## Second Appeal Brief

### Table of Contents

- (i) Real party in interest, page 2;
- (ii) Related appeals and interferences, page 2;
- (iii) Status of claims, page 2;
- (iv) Status of amendments, page 2;
- (v) Summary of claimed subject matter, page 2;
- (vi) Grounds of rejection to be reviewed on appeal, page 2;
- (vii) Argument, pages 3-15;
- (viii) Claims appendix, page 16;
- (ix) Evidence appendix, page 17;
- (x) Related proceedings appendix, page 17;
- (xi) Conclusion, page 18;
- (xii) Figures, page 19-21.

(i) Real party in interest.

The real party in interest is inventor Victor Gorelik.

(ii) Related appeals and interferences.

Brief Appeal dated 07-11-2007

(iii) Status of Claims.

Claims 1-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Buffam (US-6185316-B1) in view of Scheidt et al ("Scheidt")(US-6542608-B2).

(iv) Status of Amendments.

No amendment filed subsequent to the final rejection

(v) Summary of claimed subject matter.

Not included according to 37 CFR 41.37(c)(1): the appellant is not represented by a registered practitioner.

It is unclear why the Examiner in his non-final rejection signed 10/18/2007 states: ""the applicant's appeal brief must have the proper applicable information under headings '(v) Summary of claimed subject matter' and '(vi) Grounds of rejection to be reviewed on appeal.' regardless of whether they are represented by a registered practitioner".

(vi) Grounds of rejection to be reviewed on appeal.

Not included according to 37 CFR 41.37(c)(1): the appellant is not represented by a registered practitioner.

(vii) Argument.

Rejection under 35 U.S.C. 103(a) over U.S. Patent No. US-6185316-B1  
in view of U.S. Patent No. US-6542608-B2.

Claims 1 to 5.

The new ground of rejection is improper because neither of the patents US-6185316-B1 and US-6542608-B2 alone, nor either in view of the other, teaches or suggests the proposed solution of the primary goal of my application, i.e. protection of the privacy of the user's biometrical information in case of a server security breach (this problem is not the focus of their inventions). I achieve it (unlike US-6185316-B1 and US-6542608-B2) by using biometrics in a different way (Buffam uses encoding based on true and false biometric points, Scheidt uses randomization of digital string corresponding to biometric vector, I use shuffling of the vector and multiplying its terms by some numbers).

This statement is proved by the arguments below, which are presented as follows:

- A. Goal of invention.
- B. Buffam's method vs. my method.
- C. Scheidt's method vs. my method.
- D. Buffam's method in view of Scheidt's method vs, my method.
- E. Summary.

A. Goal of invention.

Buffam is concerned with security of the server and does not use word "privacy" in his invention at all.

Scheidt uses the word "privacy", but in different sense than I do.

In "Background of the Invention" he states: "Unfortunately, drawbacks accompany the benefits provided by electronic communication, particularly in the area of

privacy. Electronic communications may be intercepted by unintended recipients. ... One form of solution uses cryptography to provide privacy for electronic communication. ... One way to improve the security of the cryptographic scheme is to minimize the likelihood that a valid key can be stolen, calculated, or discovered.”

The first phrase of his “Summary of the Invention” is “It is therefore an object of the present invention to provide a process and apparatus for assembling keys which provides added security against compromising a communication by unauthorized entities”.

So, the goal of Scheidt’s invention is privacy of communications, the tool is cryptography, and the object of the invention is the key with added security. There is not any discussion either in Buffam’s or in Scheidt’s inventions of what could happen in the case if security of the server were compromised. This problem is not even mentioned. My application on the contrary has the goal to guarantee the privacy of the user after the security of the server is compromised. By “privacy” I mean that the real biometric information which is being used in communication cannot be restored in the case of a security breach.

Scheidt is protecting transmitted text (or similar information) and uses biometric data to create a key with added security. I am protecting biometric data which is being used in the communication. These are absolutely different goals.

The means for achieving these different goals are also different.

#### **B. Buffam’s method vs, my method.**

Buffam uses true and false image points in his encoding procedure. The claimant receives access to the server only if he or she provides correct (real) biometric data and correct plain text (password known to the user only). So, if the system lets the attacker in, he or she knows that submitted/generated biometric data is real. Using method of trial and error the attacker can restore this data and use it for purposes other than access to this particular server.

In my method the claimant (user) receives access to the server if he or she submits a shuffled array of real biometric data. The sequence of values in this shuffled array (permutation) is defined by the user during enrolment phase. This shuffled array, not the real one is the proxy of the user. So, if the system lets the attacker in, he or she knows only that he or she has submitted the correctly shuffled array of biometric data, not the real biometric data. The attacker can try to determine the original sequence of values in real biometric data array by using the method of trial and error. However, the number of possible pairs “trial original sequence – trial permutation” producing the same sequence (which was defined during enrollment and opens access to the server) may be very big, and the attacker does not have a way to find out which of these pairs contains the original sequence of values and which does not.

The fact that there are many additional solutions giving access to the server does not mean that it is easy to find even one of them. However, it means that if such a solution is found, it probably does not reveal the real biometric, and there is no way to figure out which of these solutions reveals the real biometric and which does not.

To summarize: the method of trial and error can restore real biometric data in Buffam’s invention because there is only one input which opens access to the server; however, the method of trial and error is useless for the attacker in my application, because there are too many inputs that open access to the server.

This is the main difference between Buffam’s encoding and my shuffling in this context.

Let us consider the following simplified example.

The real biometric data of the user is the set of three points e.g. (10, 11), (22, 23), (34, 35), which can be treated as two arrays: the first for x-coordinates – 10, 22, 34, and the second for y-coordinates – 11, 23, 35.

According to Claim 1 of my application the user shuffles these 2 arrays. Let us suppose, to shuffle the first array (10, 22, 34) the user applies permutation 2, 3, 1 and

receives array (22, 34, 10); to shuffle the second array (11, 23, 35) the user applies permutation 3, 1, 2 and receives array (35, 11, 23). As a result, the user has three new points: (22, 35), (34, 11), (10, 23) instead of the real ones: (10, 11), (22, 23), (34, 35).

According to Buffam's invention two new false points, e.g. (46, 47) and (58, 59), are added to the set, so there are 5 points now: (10, 11), (22, 23), (34, 35), (46, 47), (58, 59).

Claim 2 of my application states that the new sequence is calculated at the client based on information known to the user only. For this simplified example it means that both permutations – 2, 3, 1 and 3, 1, 2 – are not calculated or stored anywhere – they are determined at the client on the basis of the value of the twister known only to the user.

In Buffam's invention 2 false points are calculated on the basis of the 3 real ("true") points at this step.

According to Claim 3 of my application the arrays of biometric data may be multiplied by the sequences of numbers known at the client only, which in case of multiplying sequence e.g. {3, 10, 2, 7, 1, -1} and shuffled points (22, 35), (34, 11), (10, 23) produces points (66, 350), (68, 77), (10, -23). These three new points are the proxy of the user. They are submitted and saved at server during enrolling phase.

In Buffam's invention plain text known to the user only (e.g. "Sesame, open up!") along with 2 false points (46, 47), (58, 59) are used as input for encoding. The output is a cipher text. This cipher text along with 5 points (3 true and 2 false): (10, 11), (22, 23), (34, 35), (46, 47), (58, 59) are saved at the server during the enrollment phase.

Claim 4 of my application describes the process of verification. This process does not include decoding. Biometric data is twisted at the client and only this twisted data is submitted to the server. Information which was used to twist the data (two permutations and sequence of multipliers) is not submitted to the server. Verification is being done by comparison of twisted data submitted by the claimant against the data stored on server. If the claimant submits exact the same twisted data: (66, 350), (68, 77), (10, -23), he is the

right person and access to the account is granted. Because of the non-deterministic nature of biometric sampling, calculation of correlation coefficient may be used instead of direct comparison.

In Buffam's method claimant submits real biometric points plus plain text. These points are compared with 5 points stored at the server. Coinciding points are removed; the remaining points plus submitted plain text are used to produce the cipher text. In case if the claimant is the right person, she/he submits true points (10, 11), (22, 23), (34, 35) that will be removed from the union of five points (10, 11), (22, 23), (34, 35), (46, 47), (58, 59) stored on server. Two remaining points (46, 47), (58, 59) and the plain text are used to produce the cipher text. Because the claimant is the right person, this cipher text will coincide with the cipher text stored on server. So, access to account will be granted.

One purpose of this simplified comparison is illustration of the fact that every one of my 4 claims is different from Buffam's invention:

- claim 1 describes the shuffling of coordinates of true points instead of creating additional false points,
- claim 2 states that this shuffling is done on the basis of information known only to the user instead of calculation of new false points;
- claim 3 is additional multiplication of the coordinates instead of producing encoding key from the false points and plain text;
- claim 4 is a comparison of submitted twisted data against twisted data stored on the server instead of calculating the cipher text on the basis of submitted real biometric data and plain text.
- claim 5 is a description of means for implementing methods of claims 1-4.

The second purpose of this comparison is even more important for my arguments. It is the proof (as follows below) that, real biometric data can be restored using the data stored on the server in case of Buffam's method and cannot be restored in case of my method.

Indeed, if the attacker generates some wrong biometric data points e.g. (1, 1), (2, 2), (3, 3) or some wrong plain text e.g. "Let me in, please", the Buffam's procedure will reject the claim. However, if the attacker generates correct data points (10, 11), (22, 23), (34, 35) and correct plain text "Sesame, open up!" (randomly or using new mathematical methods – it does not matter how), the Buffam's procedure will remove these 3 points from the 5 points stored on the server, and use two remaining points and plain text to produce the cipher text, coinciding with the cipher text stored on the server. So, access to the account will be granted, and – which is the main point of my argument – the real biometric information: (10, 11), (22, 23), (34, 35) will be revealed.

In case of my method the attacker also can randomly generate correct array of biometric data: (10, 11), (22, 23), (34, 35), correct permutations: 2, 3, 1 and 3, 1, 2, and correct multiplying sequence: {3, 10, 2, 7, 1, -1}, opening access to the server. However, the attacker would not know that he/she has generated real array of biometric data. There are always other combinations ("additional solutions"), which will open access to the server as well. To make reasoning easier let us assume that there is no additional multiplication, or, the same, that all multipliers are equal to 1: {1, 1, 1, 1, 1, 1}, not {3, 10, 2, 7, 1, -1}. In this case there are 36 ( $3! \cdot 3!$ ) different solutions:

(10, 11), (22, 23), (34, 35), with permutations 2, 3, 1 and 3, 1, 2;

(22, 11), (10, 23), (34, 35), with permutations 1, 3, 2 and 3, 1, 2;

(34, 11), (10, 23), (22, 35), with permutations 3, 1, 2 and 3, 1, 2;

and so on, opening access to the server. Each of these solutions produces twisted sequences (22, 35), (34, 11), (10, 23) saved on server. Only the first of these 36 solutions has the real biometric data (10, 11), (22, 23), (34, 35).

In case of using additional multipliers (as proposed in claim 3) the number of possible solutions increases in great degree.



The fact that there are many additional solutions giving access to the server does not mean that it is easy to find even one of them. However, it means that if such a solution is found, it probably does not reveal the real biometric, and there is no way to figure out which solution reveals the real biometric and which does not.

So, the main difference between Buffam's and my methods in real-world operations is that in Buffam's method the real biometric data is collected at client and submitted to the server; in my method twisted biometric data is collected at client and submitted to the server.

As a result of this difference, if Buffam's method is used, it is possible to restore real biometric data if security of the server is broken; if my method is used, it is impossible.

#### C. Scheidt's method vs. my method.

Unlike Buffam, Scheidt randomizes a digital string corresponding to the user's biometric vector. It is also different from my shuffling of terms in the biometric vector with multiplying these terms by some numbers. My method has an advantage from the point of view of the user's privacy for the following reasons.

Scheidt describes the key composed from different splits, including the biometric split. If security of the server is compromised, then all algorithms, programs and stored data are available to the attacker. He does not know only real biometric data and secret text input of the user. So, from the point of view of trial and error method we can concentrate only on the biometric key and on the secret text input in order to figure out if it is possible to restore the real biometric vector from the randomized string.

The vector of biometric data usually has some predefined properties:

- handwritten signature is a continuous curve (or several curves);
- true image points of a fingerprint are located on ellipse-like smooth lines;

- biometric points of an iris are located on a ring,

and so on. Predefined properties of biometric points (and predefined methods of their extraction) may be used by the attacker to restore the biometric information, even if string randomization is applied.

Below are two examples of restoring real biometric data from Scheidt's randomized string on the base of predefined information, and the proof that it is impossible to restore biometric data if my method of shuffling with multiplication is used.

The first example is related to the iris biometry (fig.1- fig.4).

Let us consider the biometric vector comprised from the biometric points of an iris ring. For the purpose of illustration suppose that the following extremely simplified technology of extraction is used. All points of the ring are presented as either white or black ones. The circle located between inner and outer boundaries of the ring is drawn. This circle thus consists of the set of black and white points. The result of extraction is a biometric vector of the black points located on the circle. They are numerated by the following rule: the highest point is assigned number 1; the next clockwise point is assigned number 2, and so on.

Figure 1 illustrates the case of 12 original points with coordinates (6,11), (7,9), (9,7), (11,6), (9,5), (7,3), (6,1), (5,3), (3,5), (6), (3,7), (5,9) which correspond to the vector (6,11, 7, 9, 9, 7, 11, 6, ...), where odd terms are x-coordinates, and even terms are y-coordinates of the biometric points. Digital string may be produced from this vector using binary system: 6=0110, 11=1011, 7=0111... so the original digital string is: 0110,1011,0111,1001,1001,0111,1011,0110,... . User applies some secret permutations and receives randomized string, e.g. 0001, 0010, 0101, 1010, 0110, 1010,... . Taking into account that 0001=1, 0010=2, 0101=5, 1010=10, 0110=6, ... we see that this randomized string corresponds to vector (1,2,5,10,6,10, ... ), or (1,2), (5,10), (6,10), ... , see figure 2.

So, the attacker knows

1) randomized string 0110,1011,0111,1001,1001,0111,1011,0110,..., which corresponds to figure 2;

2) the fact that the sting before randomizing corresponds to figure 1: all 12 points with unknown coordinates are located on the same circle.

Attacker does not know either the radius of the circle or the locations of the points on the circle.

The question is whether the attacker can restore the exact coordinates of all 12 points.

The answer is “yes, it is possible to do.”

Indeed, the randomization could have been performed by a big, but limited number of permutations. The attacker can reverse all of them and receive a trial vector for each permutation in turn. The goal of the attacker is to find out the reversed permutation which transforms figure 2 into figure 1. Figure 3 shows the typical result of applying reversed permutation to points shown on figure 2. It is clear that points on figure 3 are not located on the same circle, so the attacker will continue his trials. At some moment he can have situation shown on figure 4, where all points are located on the same circle. However, the points on figure 4 are not the real biometric vector because they are not located in the strict clock-wise sequence (attacker knows the procedure how the points are extracted), so he will continue the trials. Eventually the attacker will find the needed reversed permutation and hence the real biometric vector (which is shown on figure 1). So, using only the general predefined property – the points are located on the circle – the attacker is able to find out exact coordinates of all the points in the biometric vector.

The example above is extremely simplified. It serves just as an illustration of the fact that randomization of digital string does not guarantee that real biometric data will not be restored, since predefined properties of the biometric data (and predefined methods of their extraction) are known to the attacker in advance.

My method of shuffling with multiplying completely excludes the possibility of restoring of real biometric vector. Indeed, the number of possible combinations “permutation - multipliers” transforming figure 2 into figure 1 becomes infinite even if only one multiplier is used. So the method of trial and error is not applicable. Other methods also cannot produce the result for following reasons.

Reversed actions for my method are:

- divide x-coordinates and y-coordinates of each point on figure 2 by some numbers, and then
- apply reversed permutation to resulting points.

The first step of this operation – division of x-coordinates and y-coordinates of each point – can move each point to any desired new location. So there are divisors that move all points on figure 2 to new positions, all of which are located on the same circle. The second operation just changes the order of the points on the curve. Thus, the attacker always has infinitely many solutions meeting the predefined property that all resulting points must be located on the same circle in correct order. He does not know which of these solutions is correct.

It means that the attacker cannot find the needed permutation and hence cannot restore the original biometric vector. “He can change the order of terms in the stolen twisted signature, but he does not have criteria when to stop in order to restore the real signature: he has nothing to compare with. So, the privacy of the user in my method is assured in greater degree.”

The last quote is taken from my amendment dated 04/09/2007, page 4; second paragraph. I have made this argument during prosecution of the application and the Examiner never addressed that argument. It is not clear why the Examiner was allowed to add a new ground of rejection.

The second example is related to fingerprints (fig.5- fig.7).

Let us consider the following method of obtaining biometric data. The user has his fingerprint as an image containing a set of biometric lines, fig.5. The user chooses three horizontal lines. These lines intersect the biometric lines in some number of points. These points form the original biometric vector by the following rule: the points are numerated from left to right starting from the highest line and finishing with the lowest line.

Data after randomization is shown on fig.6.

Typical result after applying trial reverse permutation is shown on fig. 7. The points are not located on three horizontal lines, so the attacker resumes his trials.

Eventually he will find permutation, which transform fig.6 into fig. 5.

Let us consider one more illustration of advantages of my method (additionally to two examples above).

If the attacker has a set of real biometric vectors, stolen from different persons, but does not know, which one of them corresponds to the particular randomized string, he or she can find this out by trial and error method. In other words attacker has one fig.2 and a set of different figures 1 (one for each person). Applying reversed permutations to fig.2 attacker will find out which one of the figures 1 corresponds to fig. 2 even without using additional predefined properties.

If my method of shuffling with multiplication is used, the attacker cannot find out correct figure 1, because there are infinitely many reversed solutions leading from the fig.2 to each of the figures 1.

#### D. Buffam's method in view of Scheidt's method vs. my method.

Buffam and Scheidt do not teach or suggest my proposed invention. As has been shown above, protection of the privacy of the user's biometrical information in case of a server security breach is not the focus of Buffam's and Scheidt's inventions and they do

not accomplish this goal. Neither does the combination, as shown in the following simplified example, fig.8-fig9.

Fig.8 depicts 12 true image points. According to Buffam's invention in view of Scheidt's invention they are transformed into 12 points shown on fig.9 using digital string randomization (as it was described above in section C). Two additional false points (shown on fig.9 as squares) along with plain text known to the user only are used as input for encoding. The output is a cipher text. This cipher text along with 14 points (12 randomized and 2 false) is saved at the server during the enrollment phase.

During the authentication attempt the real biometric points of a claimant are randomized and submitted to the server along with plain text. These points are compared with 14 points stored at the server. Coinciding points are removed; the remaining points plus submitted plain text are used to produce the cipher text. If the claimant is the right person, 12 randomized points coincide with 12 points saved on the server and will be removed from the union of 14 points. Two remaining points and the plain text are used to produce the cipher text. Because the claimant is the right person, this cipher text will coincide with the cipher text stored on server. So, access to account will be granted.

The attacker knows all 14 points saved on server, but does not know which two of them are false. Attacker can try all possible pairs and generated plain text. After some trial calculated cipher text will coincide with the cipher text saved on server. It means that 12 remaining points are randomized true image points. Now attacker the can use the method described in section C and restore real biometric information using predefined properties. In another strategy of restoring real biometrics attacker does not even need to find out the secrete plain text. He or she can try all possible 12 points out of 14 ones and apply method of section C to each of these dozen. After some trial attacker will find permutation which meets the predefined properties.

So Buffam's method in view of Scheidt's one does not guarantee that real biometric cannot be restored in case if security of the server is compromised. My method

guarantees the privacy of user for the reason described in section C: attacker has infinitely many solutions which meet predefined properties of the biometry.

#### E. Summary.

Every one of my 5 claims is different from Buffam's and Scheidt's inventions:

- claim 1 describes the shuffling of coordinates of true biometric points instead of creating additional false points (Buffam) or randomizing of a digital string corresponding to biometric vector (Scheidt);
- claim 2 states that the shuffling is done on the basis of information known only to the user instead of calculation of new false points (Buffam) or instead of randomization on the basis of user's identifying data (Scheidt);
- claim 3 is additional multiplication of the coordinates instead of producing encoding key from the false points and plain text (Buffam) or assembling the key from different splits (Scheidt);
- claim 4 is a comparison of submitted twisted data against twisted data stored on the server instead of calculating the cipher text on the basis of submitted real biometric data and plain text (Buffam) or instead of decoding transmitted text using composed biometric key (Scheidt);
- claim 5 is a description of means for implementing methods of claims 1-4.

The important difference between shuffling and randomization is the fact that shuffling allows multiplication of coordinates, in turn allowing moving the points to any positions. As shown in section C it means that for any given set of randomized points there are infinitely many solutions which meet predefined properties of the biometric vector (which is not a case for Buffam's method in view of Scheidt's method). Attacker does not have a way to figure out which of these solutions contains the real biometry, so original coordinates cannot be restored.

(viii) Claims appendix.

What I claim as my invention is:

Claim 1. A method for securely submitting biometric data from a client to a server comprising the steps of:

performing sampling of a real biometric characteristic at the client; and

shuffling arrays of real biometric characteristics in the sequence known at client only to thereby generate twisted biometric data; and

submitting the twisted biometric data from the client to the server.

Claim 2 A method according to claim 1 wherein the shuffling sequence is calculated at client on the base of the value of a secret object created at the client and known to client only.

Claim 3. A method according to claim 2 combined with the step of multiplying the arrays of biometric characteristics by the sequences of numbers fixed for each type of array and known at the client only.

Claim 4. A method according to claim 3 wherein the step of submitting of twisted biometric data is followed by the step of comparing this data against the samples of twisted biometric data saved at the server previously, in such a way, that the result of the verification and/or identification depends neither on the specific sequence in which biometric arrays were shuffled on the client, nor on the specific sequence of numbers used on the client to change the values of the arrays.

Claim 5. A system for secure use of biometric data comprising: the means for performing twisted sampling by changing the sequence of terms in biometric array and submitting data to the server, said system programmed for performing verification and/or identification of the client.



(ix) Evidence appendix.

None.

(x) Related proceedings appendix.

None.

(xi) Conclusion.

1. It is not clear why the Examiner was allowed to add a new ground of rejection: he has never addressed my argument made during the prosecution, and I have not made new arguments.

2. The new ground of rejection is improper because patents US-6185316-B1 and US-6542608-B2 would not have made obvious my proposed invention's means to solve a problem of protection of the privacy of the user's biometrical information after a server security breach (it is not the focus of their inventions):

- Buffam and Scheidt protect security of text or similar data during communication and use biometric data to create added security; I protect biometric data which is being used in the communication;

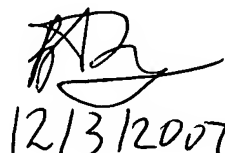
- Buffam's and Scheidt's considerations stop before the security of the server is breached; my consideration starts from this point;

- Buffam uses encoding based on true and false biometric points, Scheidt uses randomization of digital string corresponding to biometric vector, I use shuffling of the vector and multiplying its terms by some numbers.

3. Buffam's method does not protect biometric data from restoration by trial and error method. Schmidt's method also does not guarantee protection of biometric data if the attacker uses predefined properties of this data. My method explicitly and completely protects biometric data from restoration, even if the predefined properties of the biometric vector are known to the attacker. I believe this difference presents patentable novelty which the claims present and is not obvious in view of the references cited (Buffam, US-6185316-B1, Schmidt, US-6542608-B2) and the rejection made ("unpatentable").

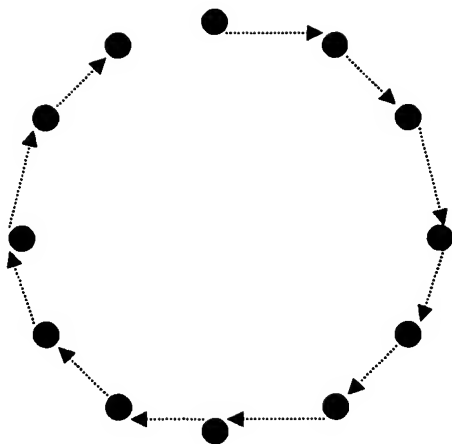
4. I am asking the Examiner and the Board to cancel rejection of the claims and to allow the application.

Inventor

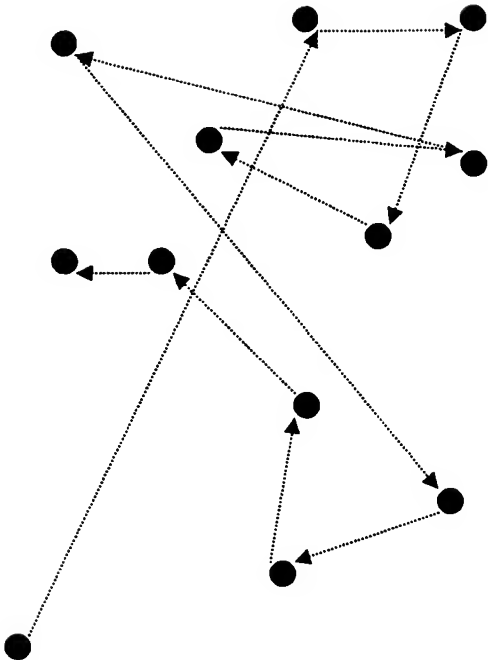


12/3/2007

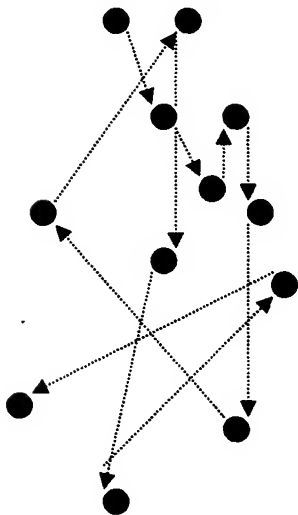
V. Gorelik.



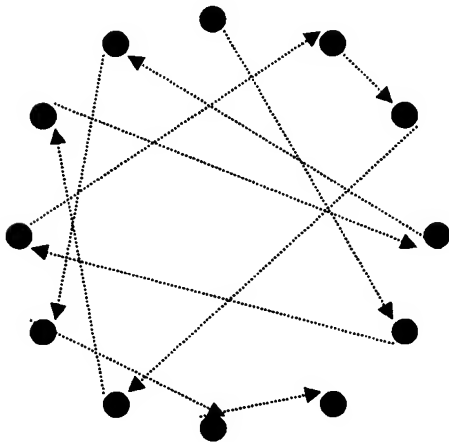
Fig, 1. Original data.



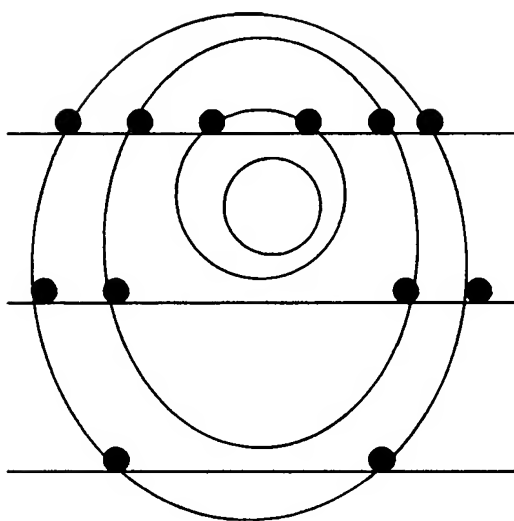
Fig, 2. Randomized data.



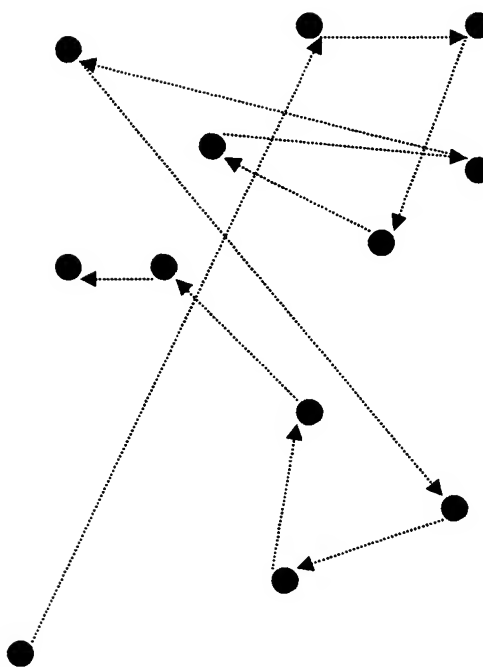
Fig, 3. Typical trial.



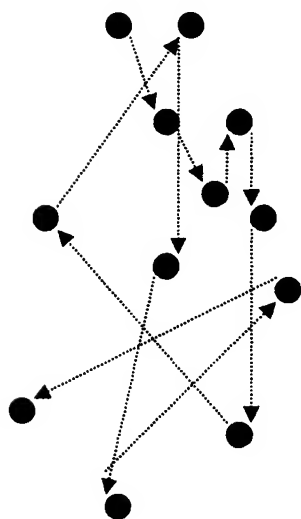
Fig, 4. Another trial.



Fig, 5. Original data.



Fig, 6. Randomized data.



Fig, 7. Typical trial.

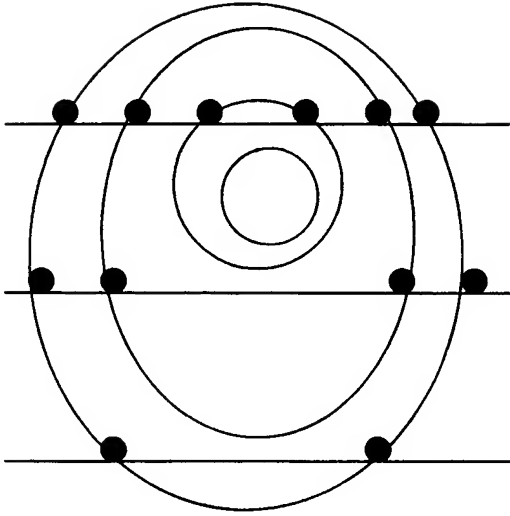


Fig. 8. Original data.

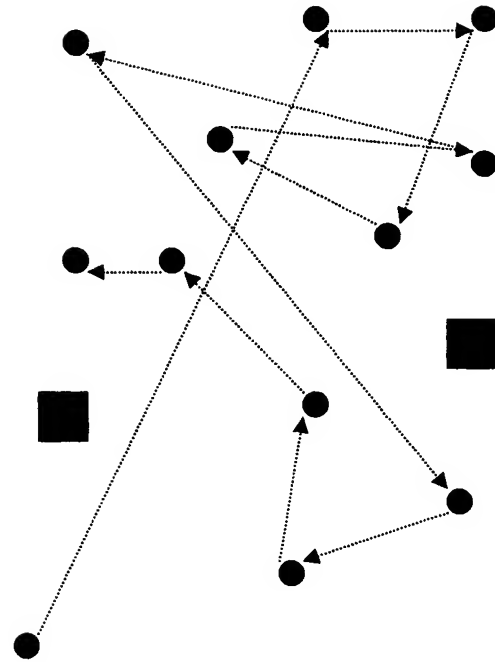


Fig. 9. Twelve true image points after randomization of the digital string and two false image points.